



Computación en la Nube

Informe Práctica 5



**Universidad
de La Laguna**

INDICE

1. Introducción.....	3
2. Backend.	3
2.1 Tecnologías usadas.....	3
2.3 Descripción API.	4
2.4 Funcionalidades implementadas.....	5
2.4.1 Subida de Archivos	5
2.4.2 Recuperar Archivos.....	6
2.4.3 Filtrado Imágenes.....	7
2.4.4 Visualización de datos.....	8
2.5 Mejoras futuras.	8
3. Frontend.....	9
3.1 Descripción.	9
3.2 Tecnologías usadas.....	9
3.3 Mejoras futuras.	9

1. Introducción.

En esta práctica se ha intentado realizar una API estática que nos permita (a través de un cliente) conectarte y poder ejecutar el código realizado en las anteriores prácticas (MPI, Secuencial y OpenMP). Para ello, usaremos la filosofía cliente/servidor. La información básica de esta práctica la podremos ver a continuación:



2. Backend.

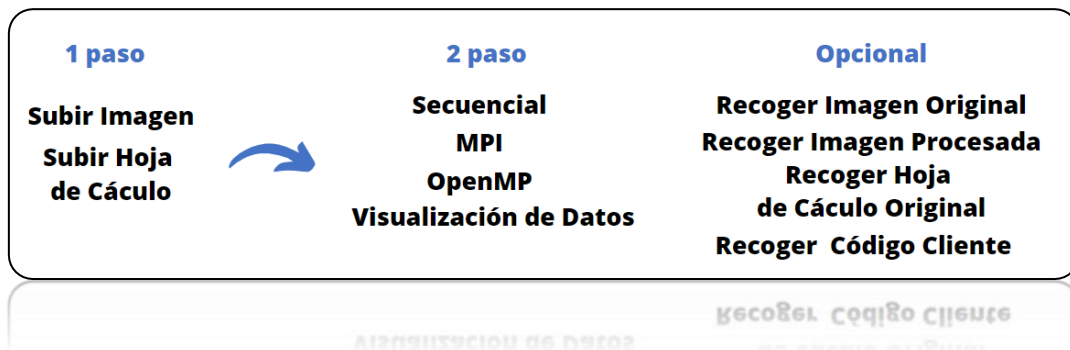
2.1 Tecnologías usadas.

En cuanto a la tecnología usada para el desarrollo del Backend la podremos ver a continuación:

- Nodejs: es un entorno de ejecución para la parte del servidor que nos permite usar Javascript como lenguaje principal. De igual forma, se han usado los módulos de Nodejs mediante el archivo "Package.json" con la información de los paquetes necesarios para el despliegue del servidor.
- Express: módulo de Nodejs que nos permitirá crear nuestra API de forma mas simple, rápida y cómoda.
- Multer: módulo de Nodejs que nos permitirá recoger y tratar los archivos que se recogerán a través de la API.
- Nodemon: módulo de Nodejs que nos permitirá emitir un servidor y que se recargue en caso de que se modifiquen alguno de los ficheros internos.

2.3 Descripción API.

En esta práctica se ha creado una API que dispone de 12 funcionalidades que nos permiten realizar diversas metodologías. Estas nos permitirán ejecutar diferentes códigos en base a una lógica predefinida. Esta sería:



Como podemos ver en la imagen anterior lo primero que debemos de realizar es una petición “POST” que permita subir el archivo que se va a usar (imagen, Excel,..). Esta petición nos devolverá una JSON explicándonos que todo ha salido correctamente y con el nuevo nombre del archivo que usaremos posteriormente. Por último, tendremos que ejecutar el algoritmo que se requiera pasándole el nombre del archivo (obtenido en el paso anterior) más los parámetros que sean necesarios para cada uno.

Como hemos mencionado anteriormente, cada función implementada dispone de unas características específicas para su uso. Estas las podremos ver en el siguiente punto o en la IP del servidor (<http://10.6.129.230>). Es esta última opción, veremos una página web con una explicación de cada funcionalidad implementada. Esta nos mostrará la siguiente información:

API Computación en la Nube

Práctica 5 Computación en la Nube

1. Subida Archivos.

1.1 Imágenes.

Elemento	Descripción
Descripción	Método que permite cargar una Imagen.
URL	/api/Upload/Img
Devuelve	<ul style="list-style-type: none">• filename: Key de la petición POST.• originalname: Nombre original del archivo.• encoding: Tipo de codificación.• mimetype: Tipo de archivo(text/jpeg,...).• destination: Lugar donde se va a almacenar.• filename: Nuevo nombre del fichero. Este deberá ser utilizado posteriormente.• path: Lugar de Almacenamiento.• size: Tamaño del archivo.

Probar

2.4 Funcionalidades implementadas.

Como hemos venido comentado se han desarrollado diversas funcionalidades que te permiten realizar diversas acciones. Estas las podremos ver a continuación.

2.4.1 Subida de Archivos

Se han implementado dos métodos que nos permite cargar una Imagen o hojas de Cálculo bajo las siguientes características:

Subida de imágenes

Elemento	Descripción
Descripción	Método que permite cargar una Imagen.
URL	/api/Upload/Img
Devuelve	<ul style="list-style-type: none">• filename: Key de la petición POST.• originalname: Nombre original del archivo.• encoding: Tipo de codificación.• mimetype: Tipo de archivo(text/jpeg,...).• destination: Lugar donde se va a almacenar.• filename: Nuevo nombre del fichero. Este deberá ser utilizado posteriormente.• path: Lugar de Almacenamiento.• size: Tamaño del archivo.

Subida de Hojas de Cálculo

Elemento	Descripción
Descripción	Método que permite cargar un archivo de tipo Hoja de Cálculo.
URL	/api/Upload/HojaCalculo
Devuelve	<ul style="list-style-type: none">• filename: Key de la petición POST.• originalname: Nombre original del archivo.• encoding: Tipo de codificación.• mimetype: Tipo de archivo(text/csv,...).• destination: Lugar donde se va a almacenar.• filename: Nuevo nombre del fichero. Este deberá ser utilizado posteriormente.• path: Lugar de Almacenamiento.• size: Tamaño del archivo.

Ambos métodos nos devolverán la información importante con respecto a dicha subida. Esta sería algo similar a la siguiente diapositiva:



2.4.2 Recuperar Archivos.

2.4.2.1 Página Web Actual.

Este método nos permitirá recoger el contenido del HTML que recogerá el cliente. Este es muy útil ya que tendríamos un ejemplo de como conectarse a nuestra API.

2.4.2.2 Imágenes Originales/Procesadas.

Este método nos permitirá recoger hojas de Cálculo/imágenes que estén almacenadas en la base de datos a través de la siguiente información:

Imágenes Original

Elemento	Descripción
Descripción	Método que permite recoger una Imagen subida previamente.
URL	/api/Get/Img/Original/:name
Parámetros	<ul style="list-style-type: none">Name: Nombre de la Imagen a recoger.

Imágenes Procesada

Elemento	Descripción
Descripción	Método que permite recoger una Imagen procesada previamente.
URL	/api/Get/Img/Final/:name
Parámetros	<ul style="list-style-type: none">Name: Nombre de la Imagen a recoger.

Hojas de Cálculo

Elemento	Descripción
Descripción	Método que permite recoger una Imagen subida previamente.
URL	/api/Get/HojaCalculo/Original/:name
Parámetros	<ul style="list-style-type: none">Name: Nombre de la HojaCalculo a recoger.

2.4.3 Filtrado Imágenes.

2.4.3.1 Filtrado Secuencial.

Este método nos permitirá ejecutar el filtrado secuencial realizado en prácticas anteriores bajo las siguientes características:

Elemento	Descripción
Descripción	Método que permite aplicar un filtrado a una imagen subida previamente. Este filtrado usa un algoritmo secuencial.
URL	/api/Secuencial/:name
Parámetros	<ul style="list-style-type: none">Name: Nombre del fichero previamente subido.
Nuevo Nombre	Secuencial-Name

2.4.3.2 Filtrado con OpenMp.

Este método nos permitirá ejecutar el filtrado en OpenMp realizado en prácticas anteriores bajo las siguientes características:

Elemento	Descripción
Descripción	Método que permite aplicar un filtrado a una imagen subida previamente. Este filtrado usa un algoritmo paralelo.
URL	/api/OpenMP/:name
Parámetros	<ul style="list-style-type: none">Name: Nombre del fichero previamente subido.
Nuevo Nombre	OpenMP-Name

2.4.3.2 Filtrado con MPI.

Este método nos permitirá ejecutar el filtrado en OpenMp realizado en prácticas anteriores bajo las siguientes características:

Elemento	Descripción
Descripción	Método que permite aplicar un filtrado a una imagen subida previamente. Este filtrado usa un algoritmo paralelo.
URL	/api/MPI/:name/:height/:width
Parámetros	<ul style="list-style-type: none">name: Nombre del fichero previamente subido.height: Altura de la imagen.width: Anchura de la imagen.
Nuevo Nombre	MPI-Name

2.4.4 Visualización de datos.

Este método permitirá ejecutar el código desarrollado en la asignatura de Análisis de datos masivos bajo las siguientes características:

Elemento	Descripción
Descripción	Método que permite ejecutar un código de representación de gráficas
URL	/api/Grafica/:dataEntry/:elementoX/:elementoY/:tipoRepresentacion/:tipoGrafica
Parámetros	<ul style="list-style-type: none">• dataEntry: Nombre del fichero previamente subido.• elementoX: Elemento del eje X.• elementoY: Elemento del eje Y.• tipoRepresentacion: En caso de agrupar el tipo de plateamiento que se quiere realizar.<ol style="list-style-type: none">1. Suma.2. Máximo.3. Mínimo.• tipoGrafica: Indicar el tipo de gráfica a devolver.<ol style="list-style-type: none">1. Gráfica de Líneas.2. Gráfica de Barras.3. Gráfica de puntos.4. Resumen.
Nuevo Nombre	<ul style="list-style-type: none">• Gráfica de Líneas: Lineas-dataEntry.• Gráfica de Barras: Barras-dataEntry.• Gráfica de Puntos: Puntos-dataEntry.• Resumen: Resumen-dataEntry.

2.5 Mejoras futuras.

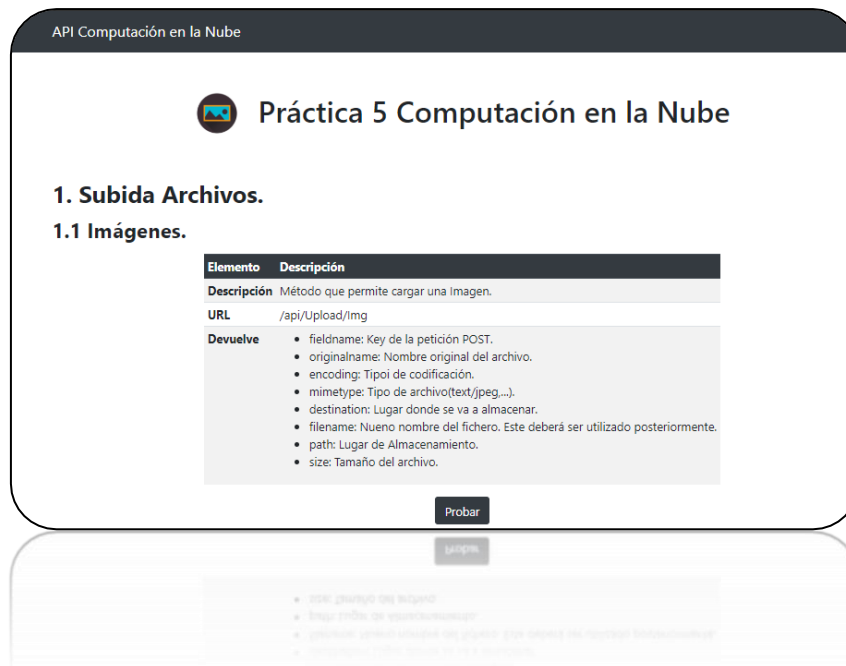
Para el desarrollo de las siguientes prácticas, se propondrán posibles mejoras para seguir avanzando en el desarrollo de nuestra API. Estas las podremos ver a continuación:

- Automatizar la creación de nuevas funcionalidades. Para ello tendremos que hacer un método que realice las siguientes funcionalidades:
 - Modificar el fichero server.js añadiéndole la nueva funcionalidad.
 - Añadirle al Makefile la nueva forma de compilar el nuevo código.
 - Añadir el código dentro de la carpeta Métodos y compilarlo la primera vez.
- Crear un método que nos devuelva un JSON la estructura del formulario para un método específico.

3. Frontend.

3.1 Descripción.

Para este apartado, se ha creado una interfaz gráfica que permita conectarse a todos los apartados anteriormente citados. Para ello, tendremos que conectarnos a la IP “10.6.129.230” y podremos ver la siguiente página:



3.2 Tecnologías usadas.

Para este apartado se han utilizado las tecnologías que veremos a continuación:

- **Html, Css y Javascript:** tecnologías básicas para el despliegue de una interfaz web.
- **JQuery:** es una librería de JavaScript (JavaScript es un lenguaje de programación muy usado en desarrollo web). Esta librería de código abierto y nos permite simplificar la tarea de programar en JavaScript.

3.3 Mejoras futuras.

Para el desarrollo de las siguientes prácticas, se propondrán posibles mejoras para seguir avanzando en el desarrollo de nuestro Frontend. Estas las podremos ver a continuación:

- Cambiar la estética a una más familiar y simplificado.
- Automatizar el HTML creado mediante los procedimientos utilizados en el Backend.
- Separar el Cliente y servidor.